


Nama: Ibnu Fajar Setiawan NIM: 065002000006	 Praktikum Data Warehouse	MODUL 8 Nama Dosen: Ir. Teddy Siswanto, MMSi
Hari/Tanggal: Hari, 30/05/2022		Nama Asisten Labratorium: 1. Azhar Rizki Zulma 065001900001 2. Nadiya Amanda Rizkania 064001900003

Persiapan Proyek Akhir 2

1. Teori Singkat

Data warehouse adalah jenis sistem manajemen data yang dirancang untuk memungkinkan dan mendukung kegiatan business intelligence (BI), terutama analitik. Gudang data semata-mata dimaksudkan untuk melakukan kueri dan analisis dan sering berisi sejumlah besar data historis. Data dalam gudang data biasanya berasal dari berbagai sumber seperti file log aplikasi dan aplikasi transaksi. Gudang data memusatkan dan mengkonsolidasikan sejumlah besar data dari berbagai sumber. Kemampuan analitisnya memungkinkan organisasi untuk memperoleh wawasan bisnis yang berharga dari data mereka untuk meningkatkan pengambilan keputusan. Seiring waktu, ia membangun catatan sejarah yang dapat sangat berharga bagi para ilmuwan data dan analis bisnis. Karena kemampuan ini, gudang data dapat dianggap sebagai "sumber kebenaran tunggal" organisasi.

2. Alat dan Bahan

Hardware : Laptop/PC

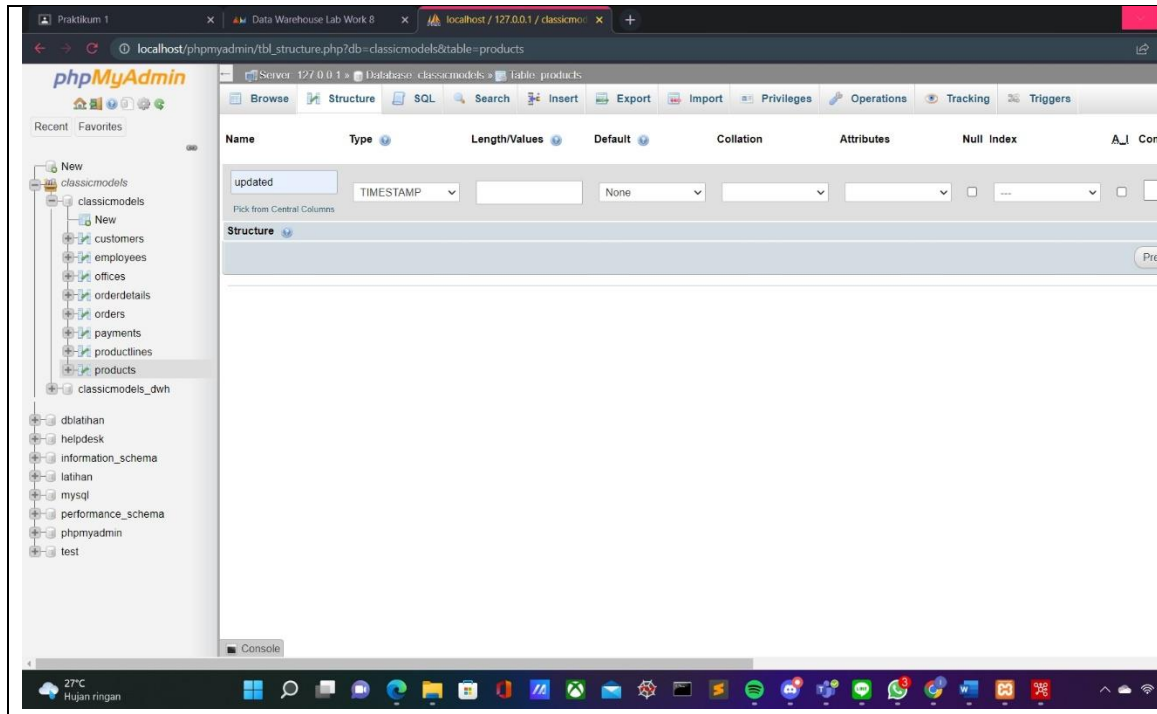
Software : Spoon Pentaho from Hitachi Vantara



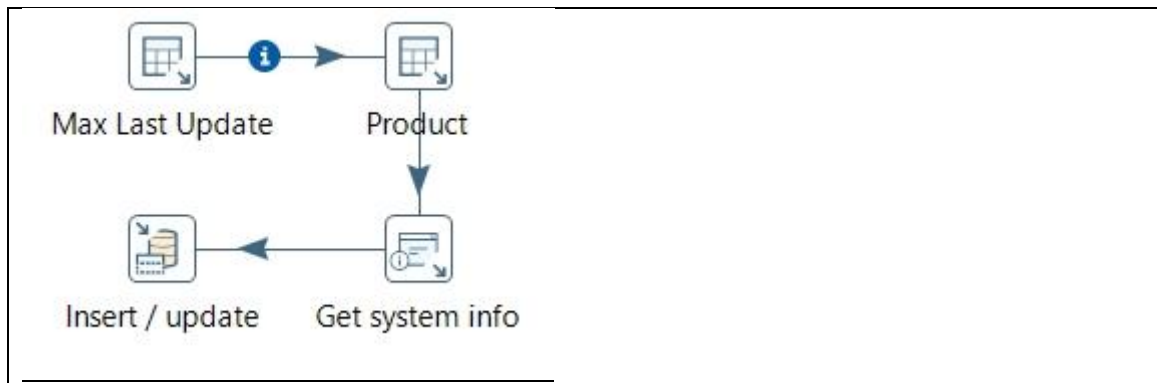
3. Elemen Kompetensi

a. Latihan pertama – Dim Product Transformation

1. Tambahkan kolom baru bernama updated dengan tipe data timestamp pada tabel products dalam database classicmodels seperti pada gambar dibawah ini di MySQL.



2. Struktur Dim Product



3. Max Last Update – Table input



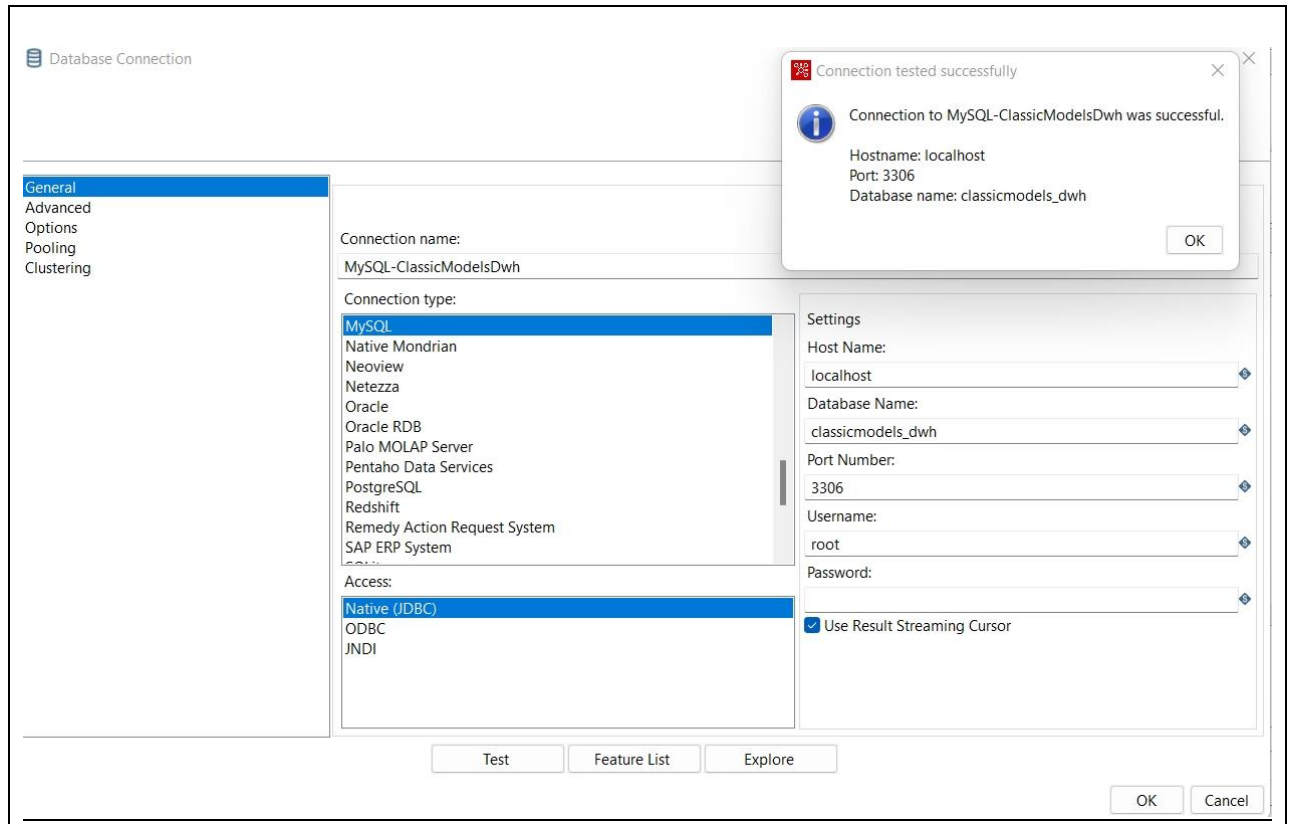


Table input

Step name: Max Last Update

Connection: MySQL-ClassicModelsDwh Edit... New... Wizard...

SQL Get SQL select statement...

```
SELECT  
COALESCE(  
    MAX(last_update),  
    '1970-01-01 00:00:00'  
) max_last_update  
FROM dim_product
```

Line 1 Column 0

Store column info in step meta ☐

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size: 0

Help OK Preview Cancel

4. Product – Table Input.



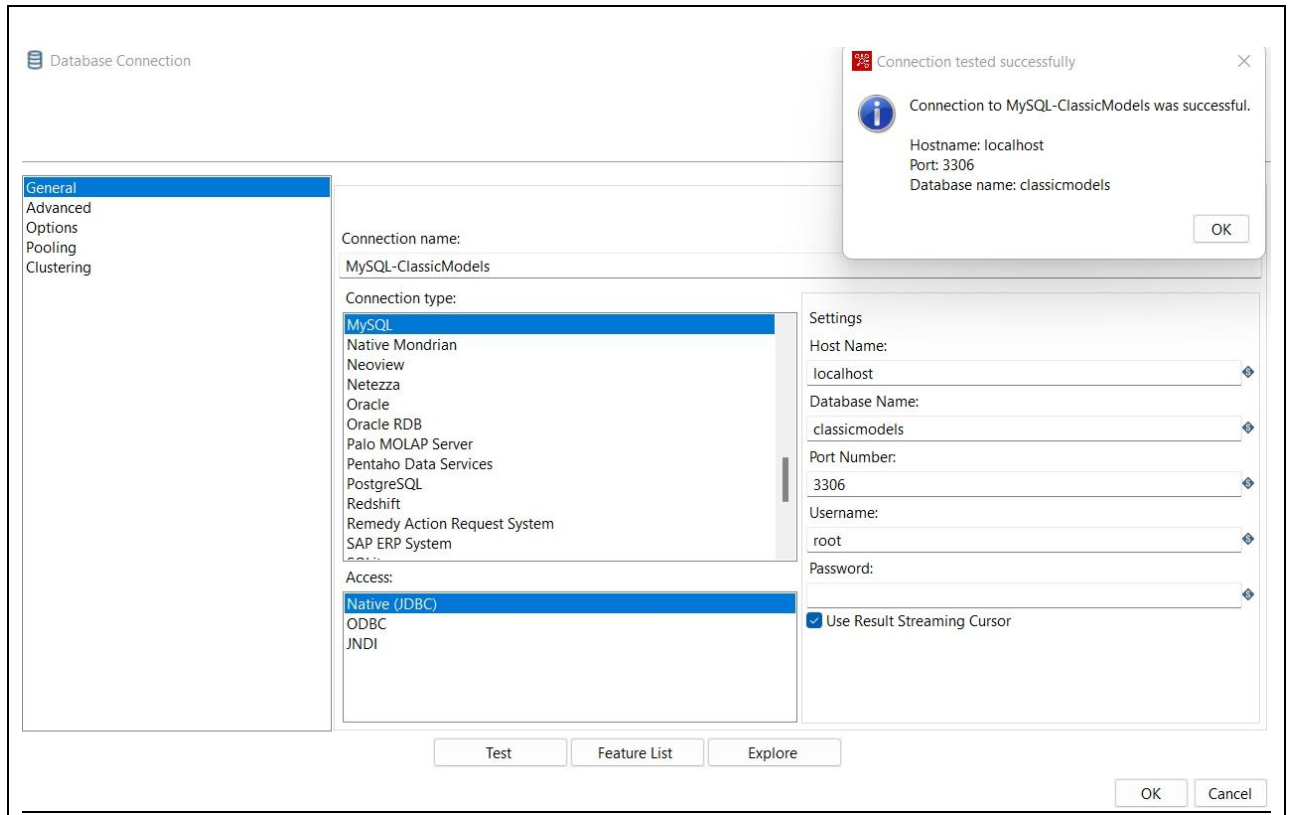


Table input

Step name: Product

Connection: MySQL-ClassicModels

Buttons: Edit... New... Wizard... Get SQL select statement...

SQL

```
SELECT productCode AS product_code,  
productName AS product_name,  
productLine AS product_line,  
productVendor AS product_vendor  
FROM products  
WHERE updated > ?
```

Line 1 Column 0

Store column info in step meta ☐

Enable lazy conversion ☐

Replace variables in script? ☒

Insert data from step: Max Last Update


Execute for each row? ☐

Limit size: 0

Buttons: ? Help OK Preview Cancel

5. Get System Info.




 Get system info

Step name

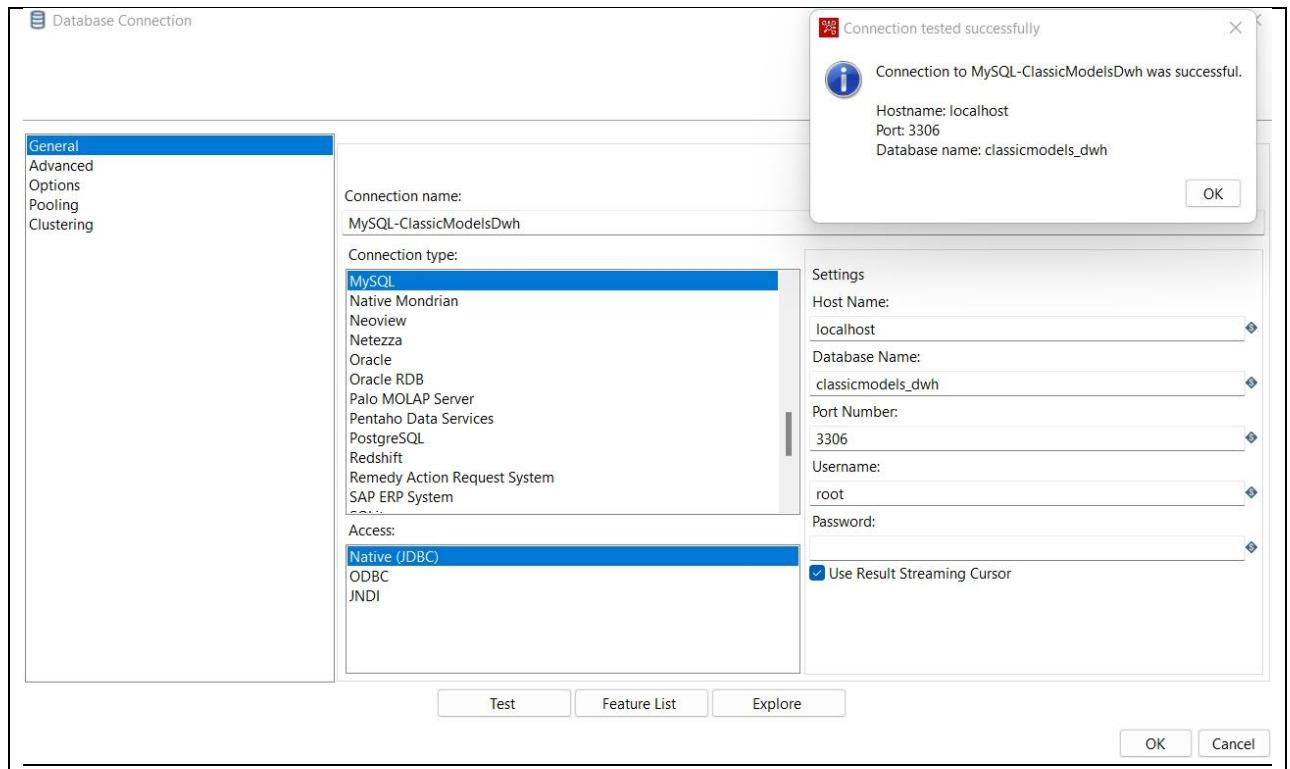
Fields:

#	Name	Type
1	last_update	system date (fixed)

 Help

6. Insert / Update.





Insert / update

Step name: Insert / update

Connection: MySQL-ClassicModelsDwh [Edit...] [New...] [Wizard...]

Target schema: classicmodels_dwh [Browse...]

Target table: dim_product [Browse...]

Commit size: 100

Don't perform any updates: ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	product_code	=	product_code	

[Get fields]

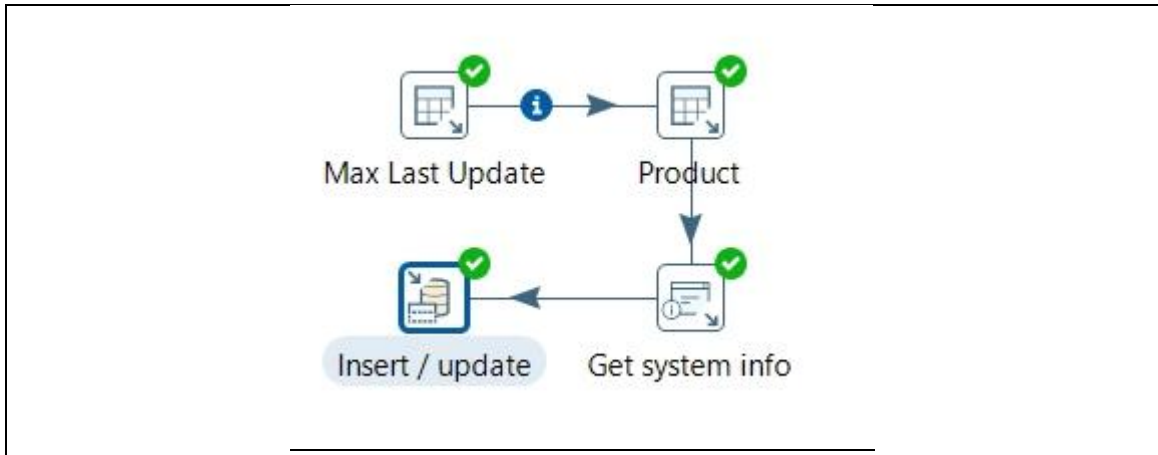
Update fields:

#	Table field	Stream field	Update
1	product_code	product_code	Y
2	product_name	product_name	Y
3	product_line	product_line	Y
4	product_vendor	product_vendor	Y
5	last_update	last_update	Y

[Get update fields] [Edit mapping]

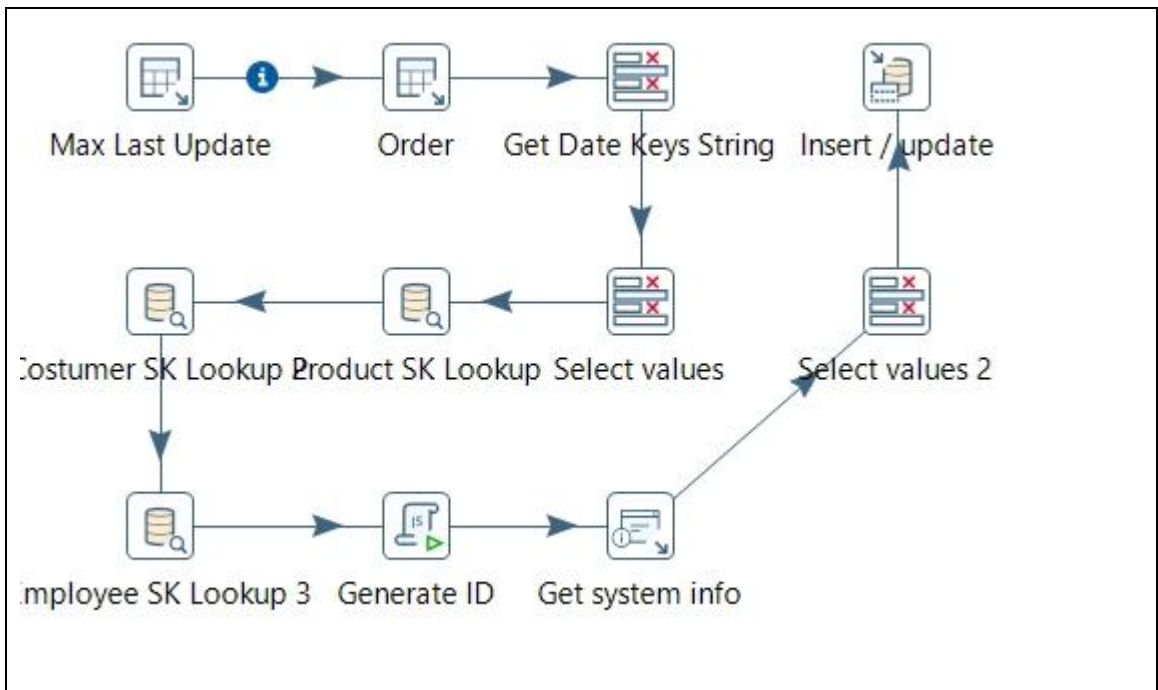
[?] Help [OK] [Cancel] [SQL]

7. Output ketika dijalankan/dirunning.



b. Latihan Kedua – Fact Order Transformation

1. Struktur Fact Order.



2. Max Last Update – Table input



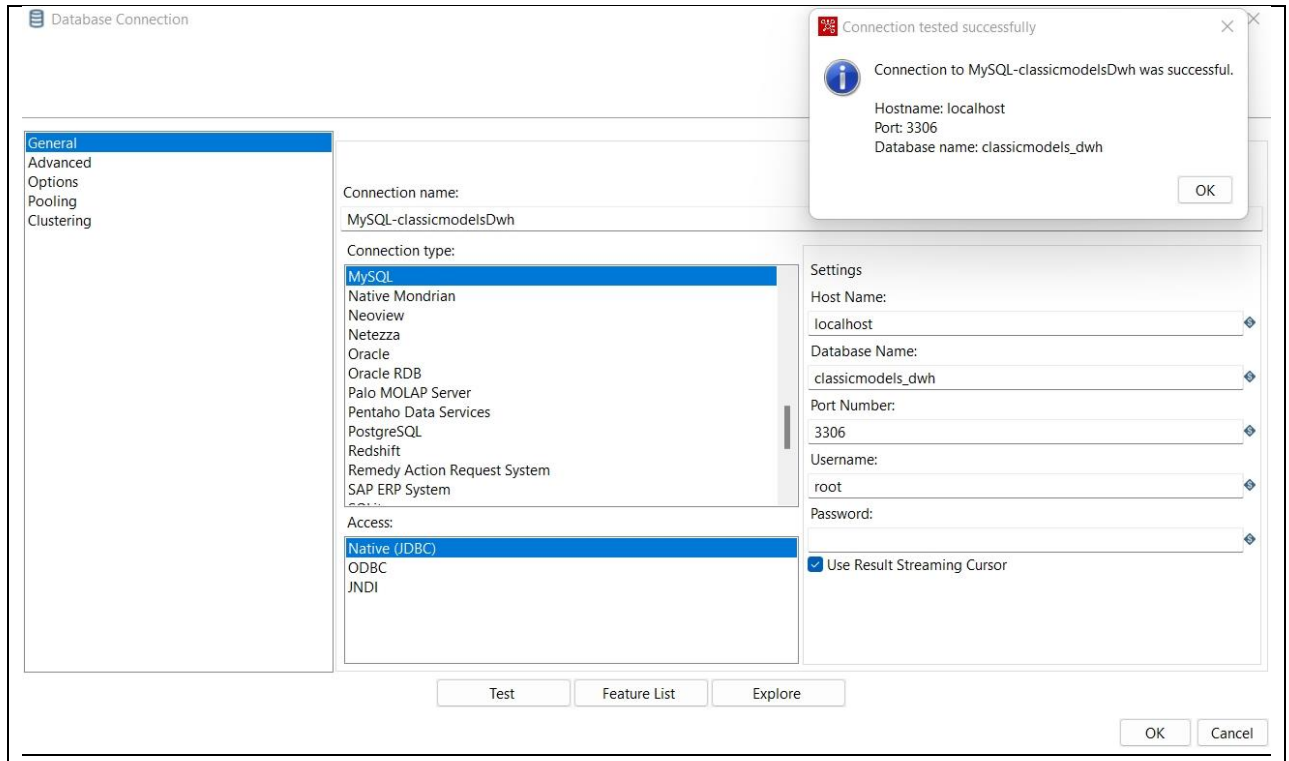


Table input

Step name: Max Last Update

Connection: MySQL-classicmodelsDwh [Edit... New... Wizard...]

SQL

Get SQL select statement...

```
SELECT  
COALESCE(MAX(last_update), '1970-01-01')  
FROM fact_order
```

Line 1 Column 0

Store column info in step meta ☐

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

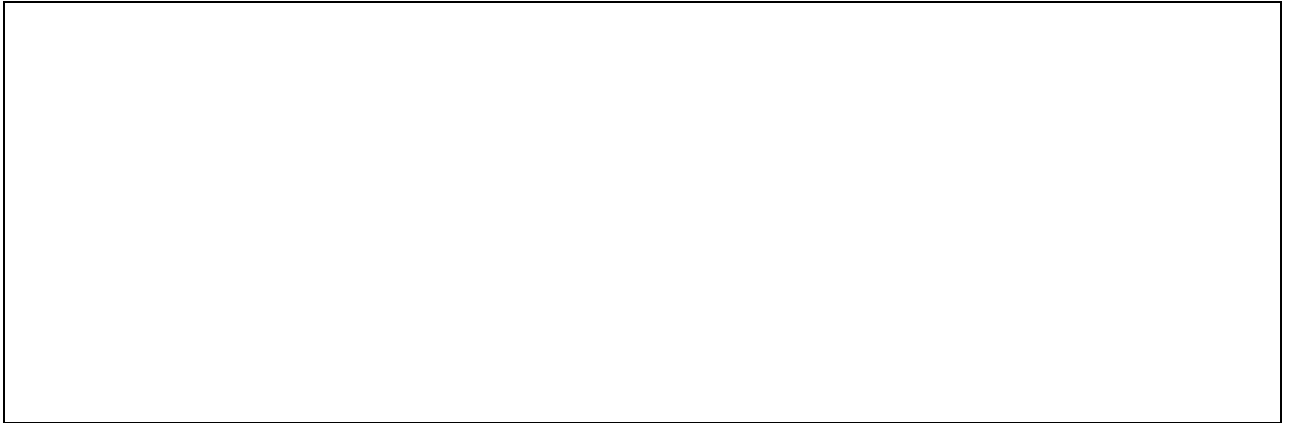
Execute for each row? ☐

Limit size: 0

? Help OK Preview Cancel

3. Order – Table input.





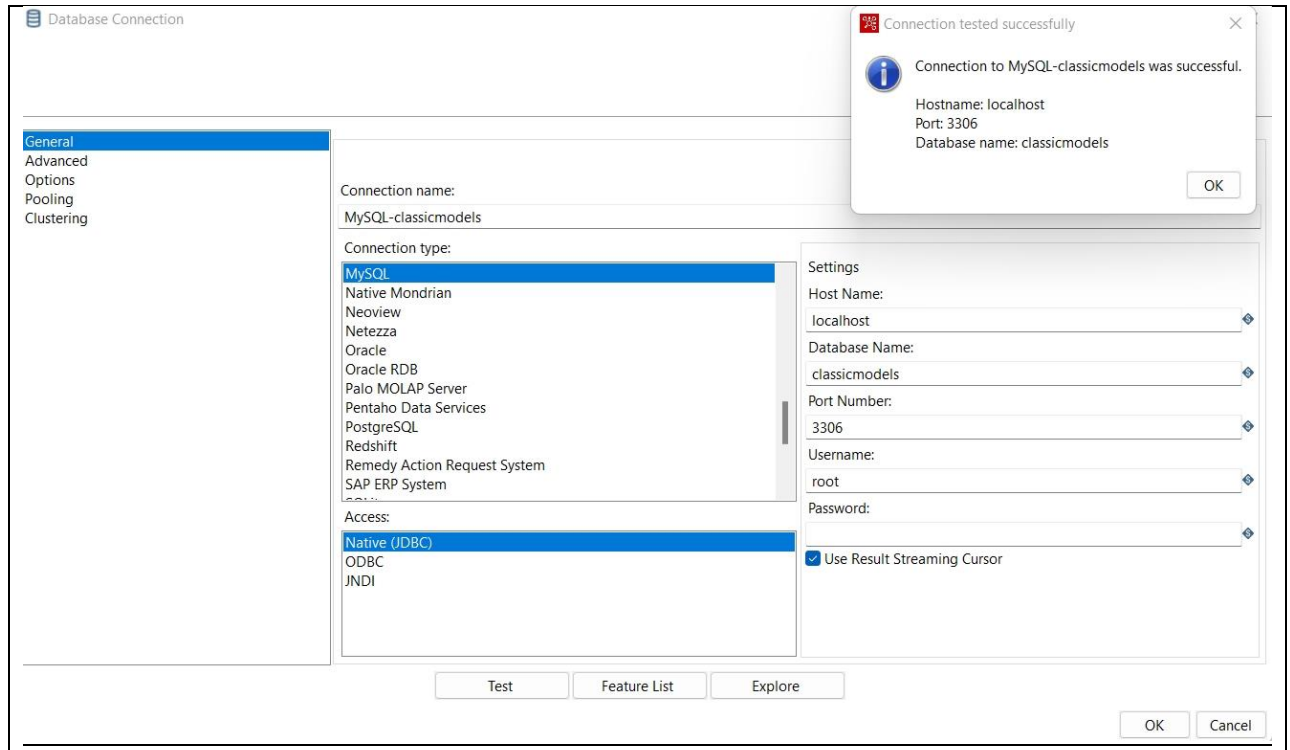


Table input

Step name: Order

Connection: MySQL-classicmodels [Edit... New... Wizard...]

SQL [Get SQL select statement...]

```
SELECT od.orderLineNumber AS order_line_number,
od.orderNumber AS order_number,
od.productCode AS product_code,
od.quantityOrdered AS quantity_ordered,
od.priceEach AS price_each,
od.quantityOrdered * od.priceEach AS price_total,
o.orderDate AS order_date,
o.requiredDate AS required_date,
o.shippedDate AS shipped_date,
e.employeeNumber AS employee_number,
o.customerNumber AS customer_number
FROM orderdetails od
LEFT JOIN orders o ON o.orderNumber = od.orderNumber
LEFT JOIN customers c ON c.customerNumber = o.customerNumber
LEFT JOIN employees e ON e.employeeNumber = c.salesRepEmployeeNumber
WHERE o.orderDate > ?
ORDER BY od.orderNumber, od.orderLineNumber
```

Line 1 Column 0

Store column info in step meta ☐

Enable lazy conversion ☐

Replace variables in script? ☒

Insert data from step: Max Last Update

Execute for each row? ☐

Limit size: 0

[?] Help [OK] [Preview] [Cancel]

4. Get Data Keys String – Select values.



Select values

Step name Get Date Keys String

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	order_line_number			
2	order_number			
3	product_code			
4	quantity_ordered			
5	price_each			
6	price_total			
7	order_date			
8	required_date			
9	shipped_date			
1..	employee_number			
1..	customer_number			

Get fields to select
Edit Mapping

Include unspecified fields, ordered by name ☐

Help OK Cancel

Select values

Step name Get Date Keys String

Select & Alter Remove Meta-data

Fields to alter the meta-data for :

#	Fieldname	Rename to	Type	Length	Precision	Binary to Normal?
1	order_date		String			N
2	required_date		String			N
3	shipped_date		String			N

Get fields to change

Help OK Cancel

5. Select values

Select values

Step name Select values

Select & Alter Remove Meta-data

Fields :

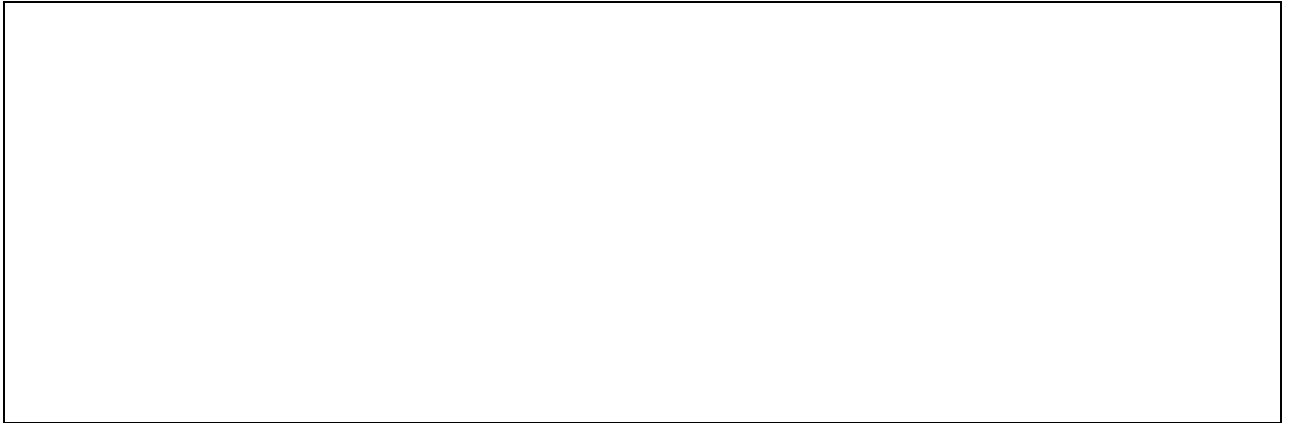
#	Fieldname	Rename to	Length	Precision
1	order_line_number			
2	order_number			
3	product_code			
4	quantity_ordered			
5	price_each			
6	price_total			
7	order_date			
8	required_date			
9	shipped_date			
1..	employee_number			
1..	customer_number			

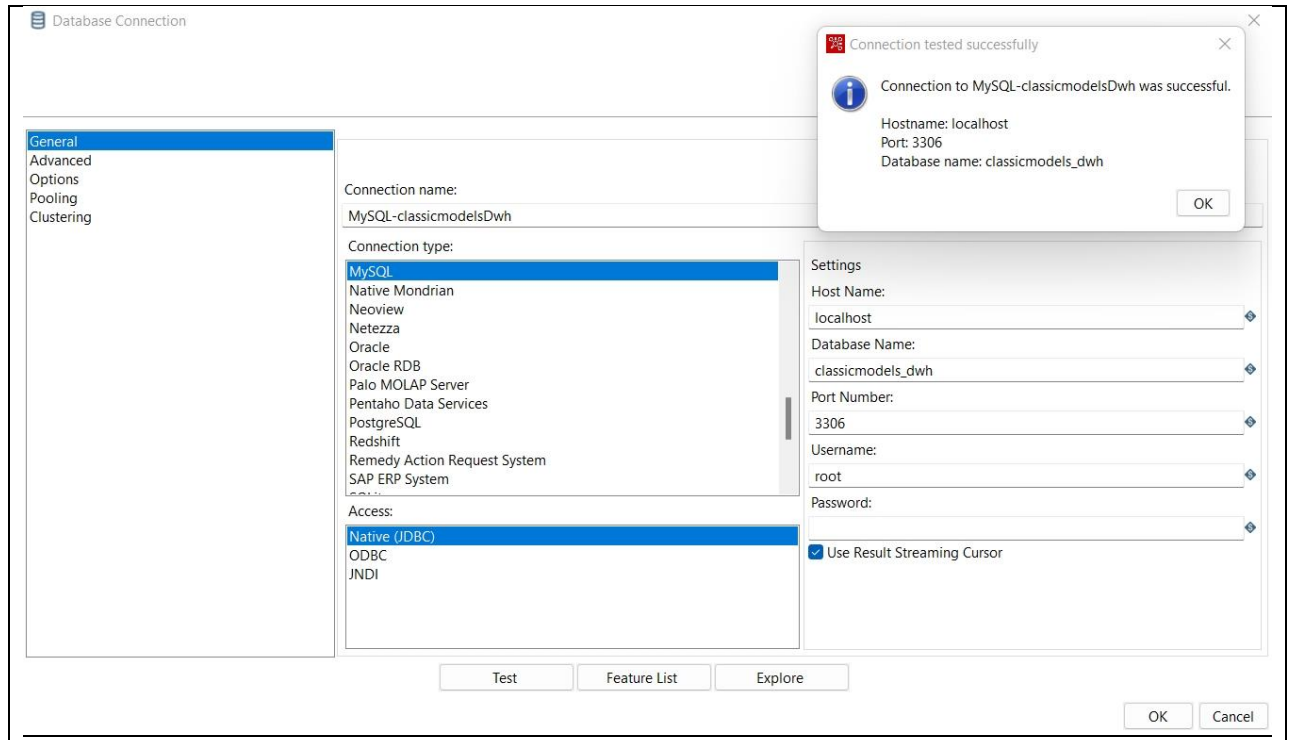
Get fields to select
Edit Mapping

Include unspecified fields, ordered by name ☐

? Help OK Cancel

6. Product SK Lookup – Database Lookup.





Database lookup

Step name: Product SK Lookup

Connection: MySQL-classicmodelsDwh Edit... New... Wizard...

Lookup schema: classicmodels_dwh Browse...

Lookup table: dim_product Browse...

Enable cache? ☐

Cache size in rows (0=cache): 0

Load all data from table ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	product_code	=	product_code	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	sk	product_sk		Integer

Do not pass the row if the lookup fails ☐

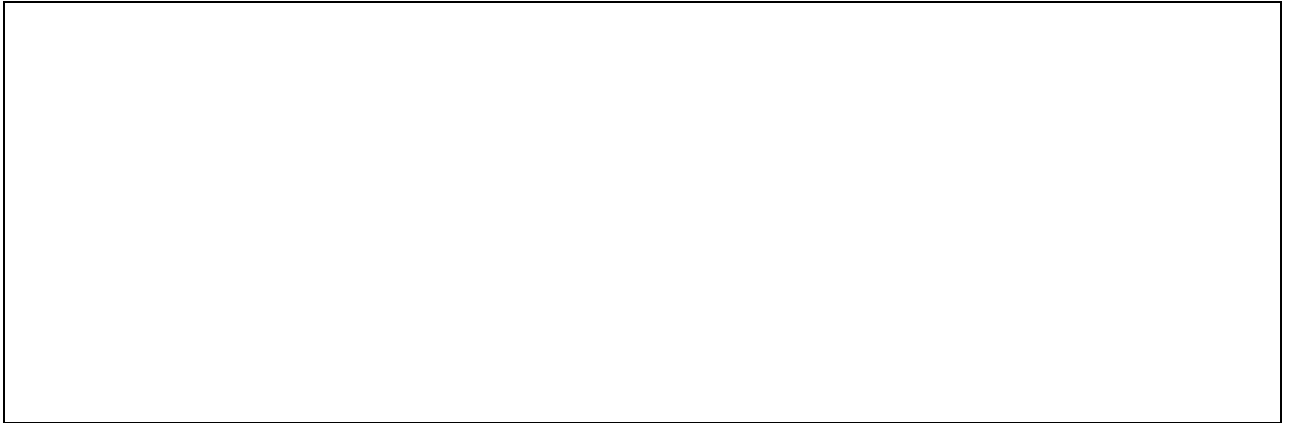
Fail on multiple results? ☐

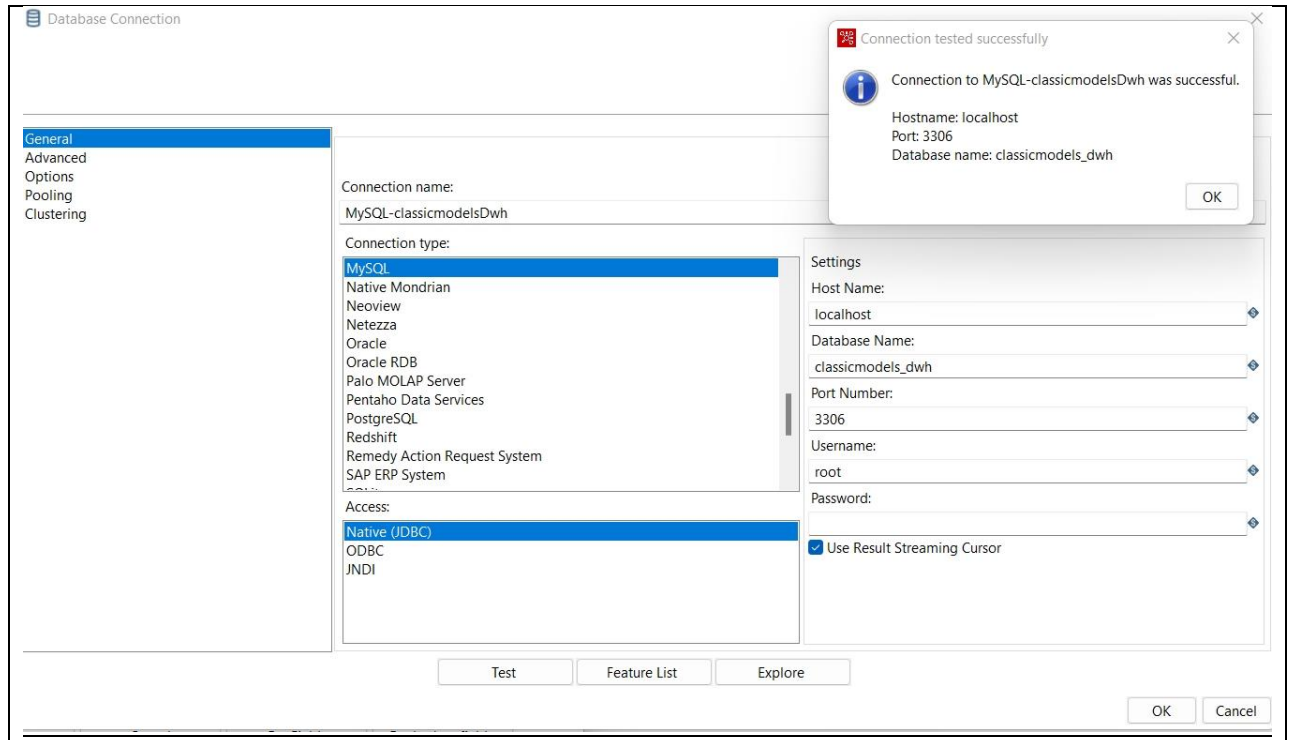
Order by:

Help OK Cancel Get Fields Get lookup fields

7. Costumer SK Lookup – Database Lookup.







Database lookup

Step name: Costumer SK Lookup 2

Connection: MySQL-classicmodelsDwh Edit... New... Wizard...

Lookup schema: classicmodels_dwh Browse...

Lookup table: dim_customer Browse...

Enable cache? ☐

Cache size in rows (0=cache): 0

Load all data from table ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	customer_number	=	customer_number	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	sk	customer_sk		Integer

Do not pass the row if the lookup fails ☐

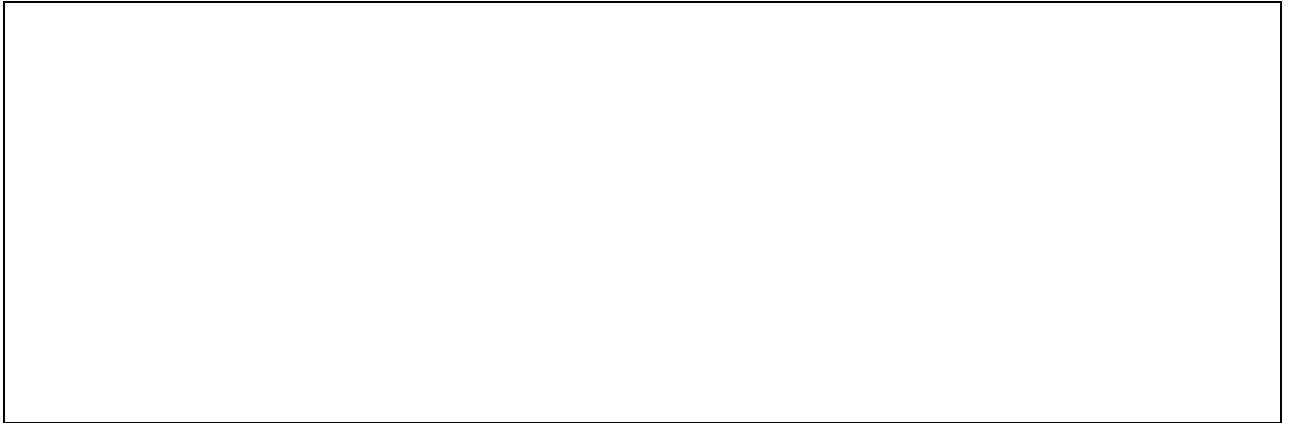
Fail on multiple results? ☐

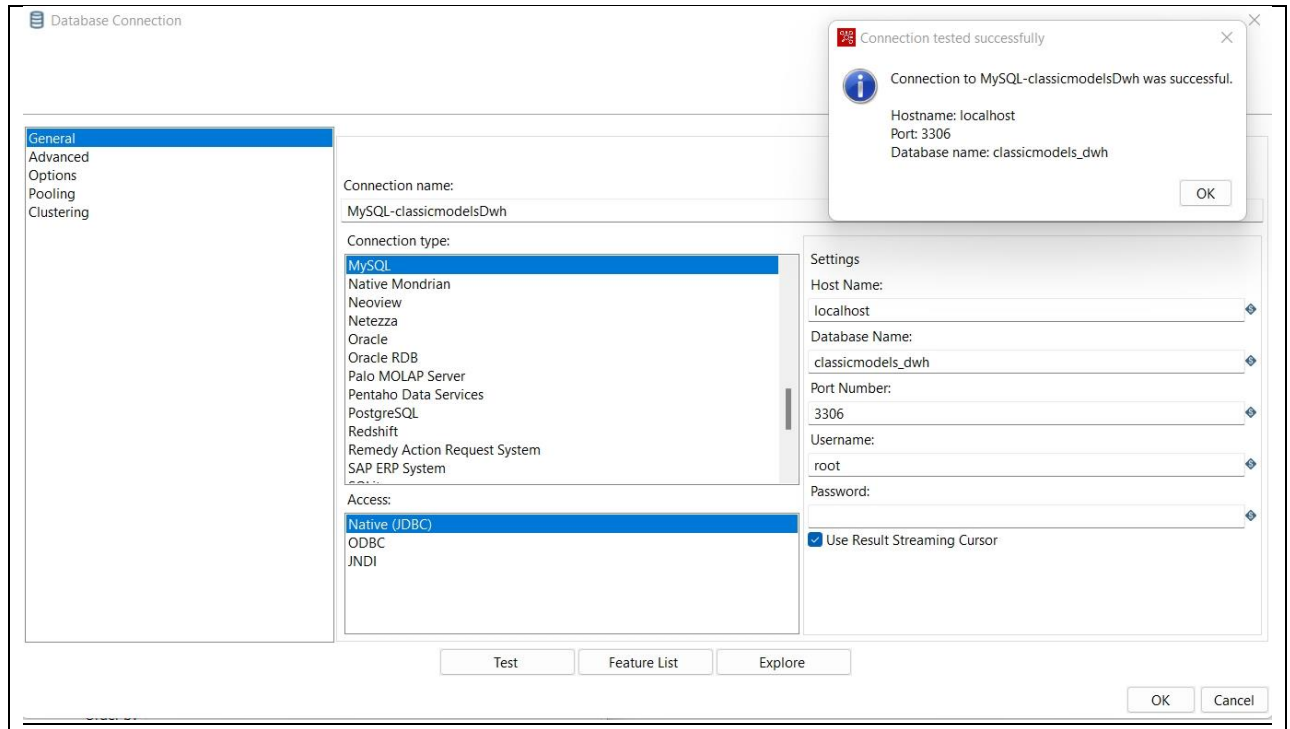
Order by:

Help OK Cancel Get Fields Get lookup fields

8. Employee SK Lookup – Database Lookup.







Database lookup

Step name: Employee SK Lookup 3

Connection: MySQL-classicmodelsDwh [Edit...] [New...] [Wizard...]

Lookup schema: classicmodels_dwh [Browse...]

Lookup table: dim_employee [Browse...]

Enable cache? ☐

Cache size in rows (0=cache): 0

Load all data from table ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	employee_number	=	employee_number	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	sk	employee_sk		Integer

Do not pass the row if the lookup fails ☐

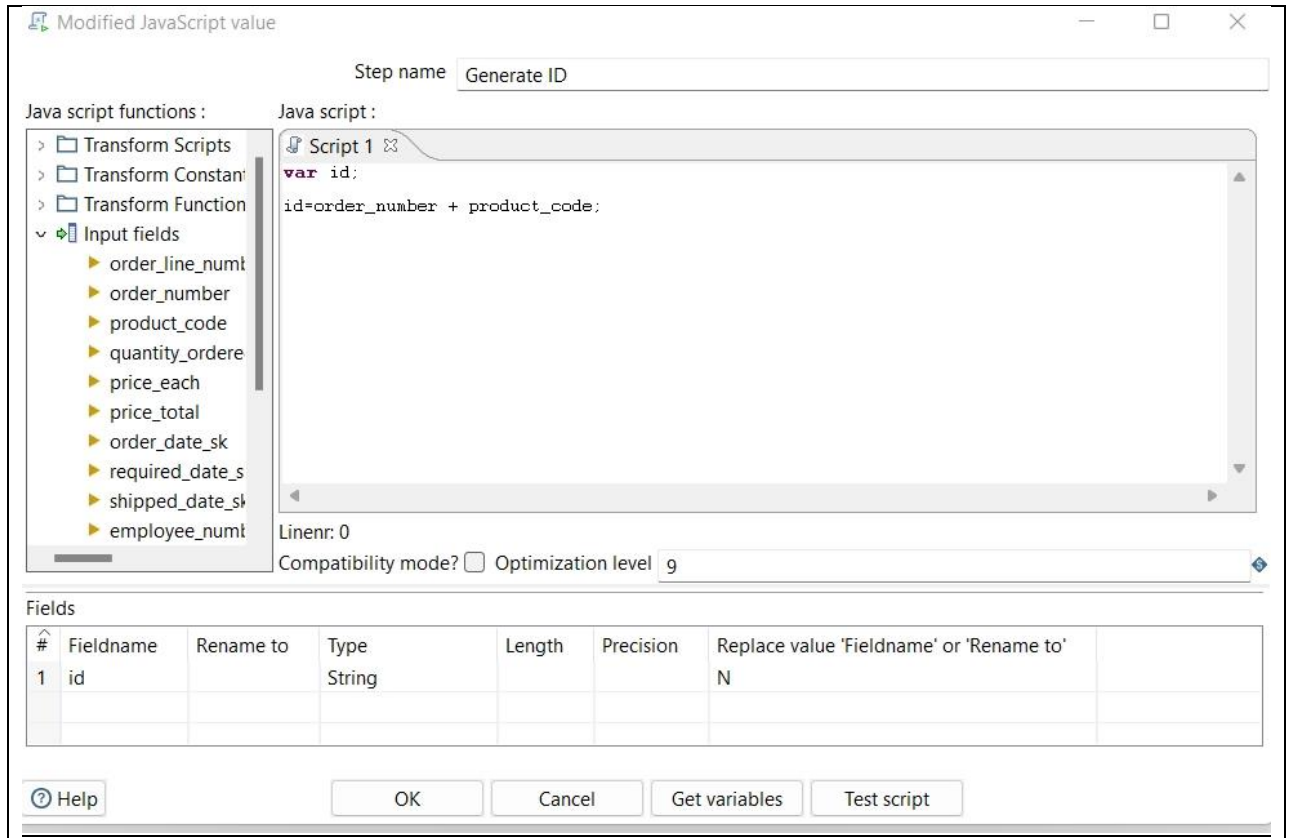
Fail on multiple results? ☐

Order by:

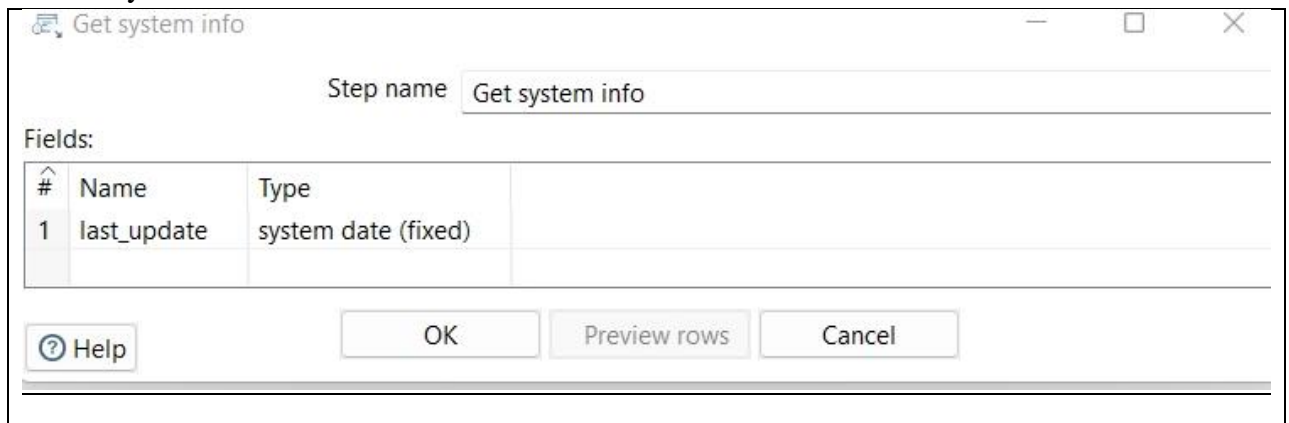
[?] Help [OK] [Cancel] [Get Fields] [Get lookup fields]

9. Generate ID – Modified JavaScript value.





10. Get System Info.



11. Select values 2.



Select values

Step name: Select values 2

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	id			
2	order_line_number			
3	order_number			
4	product_sk			
5	quantity_ordered			
6	price_each			
7	price_total			
8	order_date_sk			
9	required_date_sk			
1..	shipped_date_sk			
1..	customer_sk			
1..	employee_sk			
1..	last_update			

Get fields to select

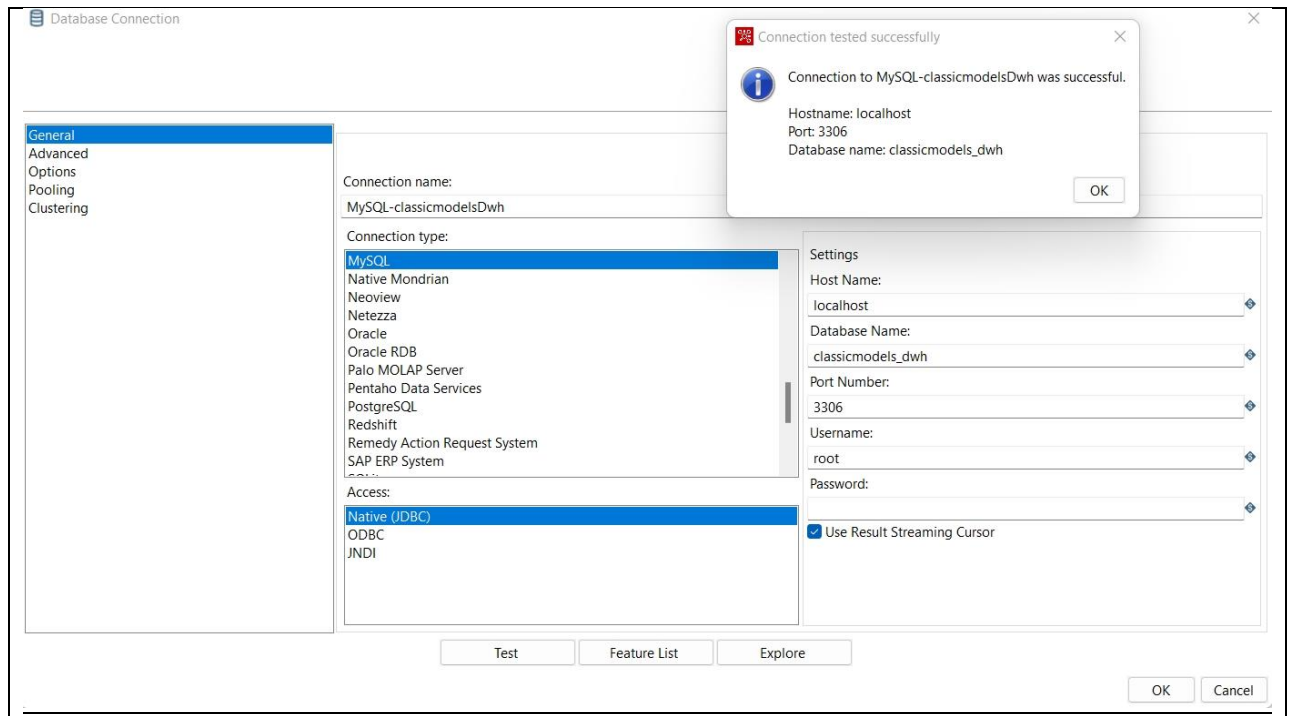
Edit Mapping

Include unspecified fields, ordered by name ☐

Help OK Cancel

12. Insert / Update.





Insert / update

Step name

Insert / update

Connection

MySQL-classicmodelsDwh

Edit...

New...

Wizard...

Target schema

classicmodels_dwh

Browse...

Target table

fact_order

Browse...

Commit size

100

Don't perform any updates:

☐

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	id	=	id	

Get fields

Update fields:

#	Table field	Stream field	Update
1	id	id	Y
2	order_line_number	order_line_number	Y
3	order_number	order_number	Y
4	product_sk	product_sk	Y
5	quantity_ordered	quantity_ordered	Y
6	price_each	price_each	Y
7	price_total	price_total	Y
8	order_date_sk	order_date_sk	Y
9	required_date_sk	required_date_sk	Y
1..	shipped_date_sk	shipped_date_sk	Y
1..	customer_sk	customer_sk	Y
1..	employee_sk	employee_sk	Y
1..	last_update	last_update	Y

Get update fields

Edit mapping

Help

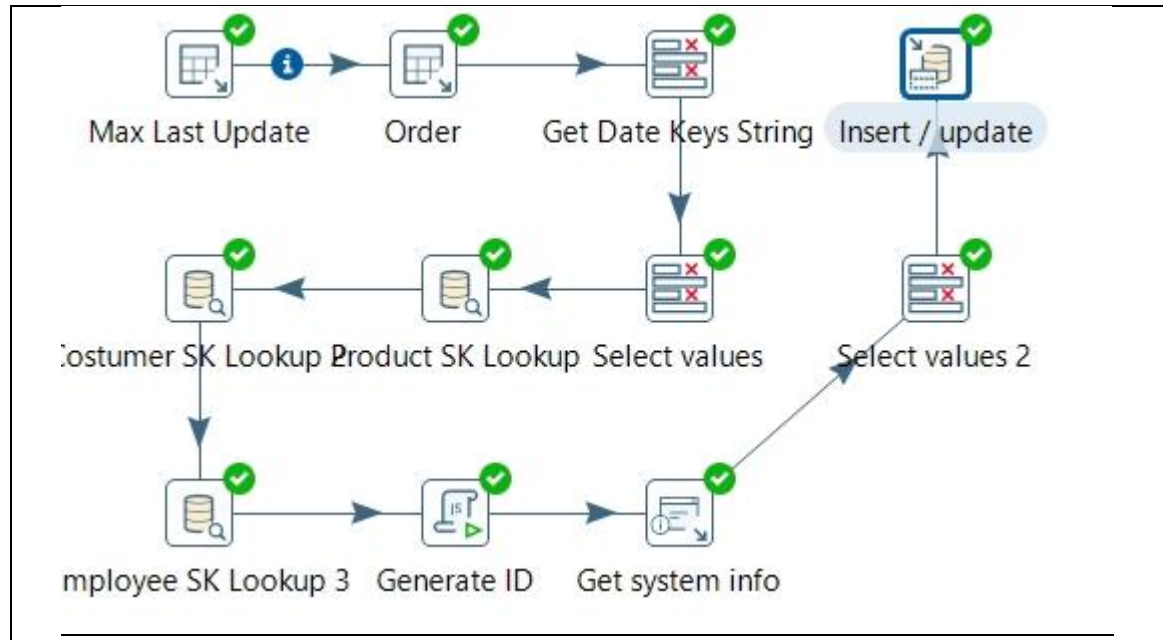
OK

Cancel

SQL



12. Output.



c. Latihan Ketiga – Order Job

1. Buatlah Job baru dan simpan dengan nama Order. Lalu buatlah struktur seperti gambar dibawah ini.



2. Dim Product – Transformation.

Transformation

Entry Name:
Dim Product

Transformation:
D:/file kuliah smstr4/fileprakdatawarehouse/prak8_1.ktr Browse...

Options Logging Arguments Parameters

Run configuration:
Pentaho local

Execution

- ☐ Execute every input row
- ☐ Clear results rows before execution
- ☐ Clear results files before execution
- ☒ Wait for remote transformation to complete
- ☐ Follow local abort to remote transformation
- ☐ Suppress result data from remote transformation

Help OK Cancel

3. Dim Employee – Transformation.



Transformation

Entry Name:
Dim Employee

Transformation:
D:/file kuliah smstr4/filepraktikdatawarehouse/prak 7/prak7_3.ktr Browse...

Options Logging Arguments Parameters

Run configuration:
Pentaho local

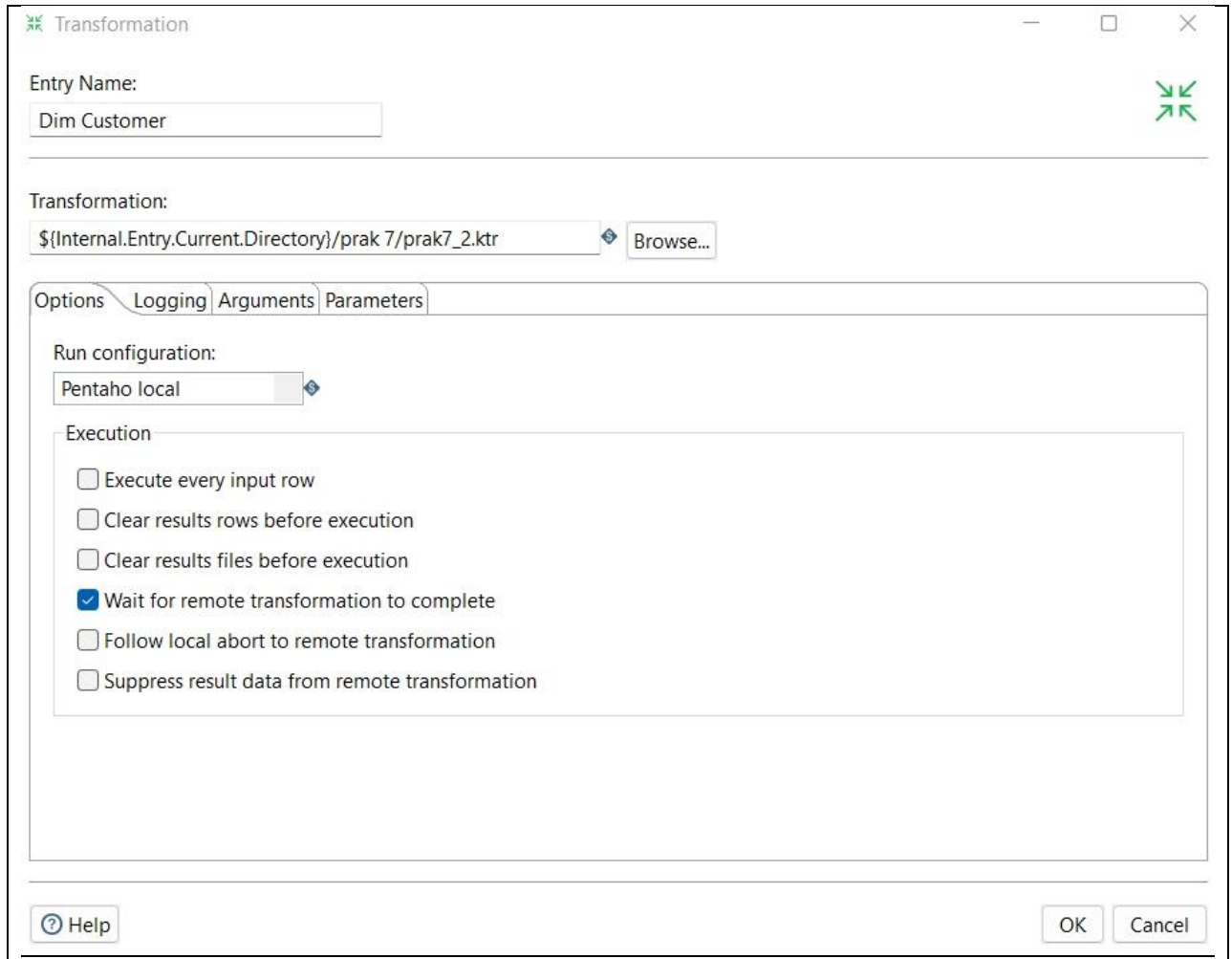
Execution

- ☐ Execute every input row
- ☐ Clear results rows before execution
- ☐ Clear results files before execution
- ☒ Wait for remote transformation to complete
- ☐ Follow local abort to remote transformation
- ☐ Suppress result data from remote transformation

Help OK Cancel

4. Dim Customer – Transformation.



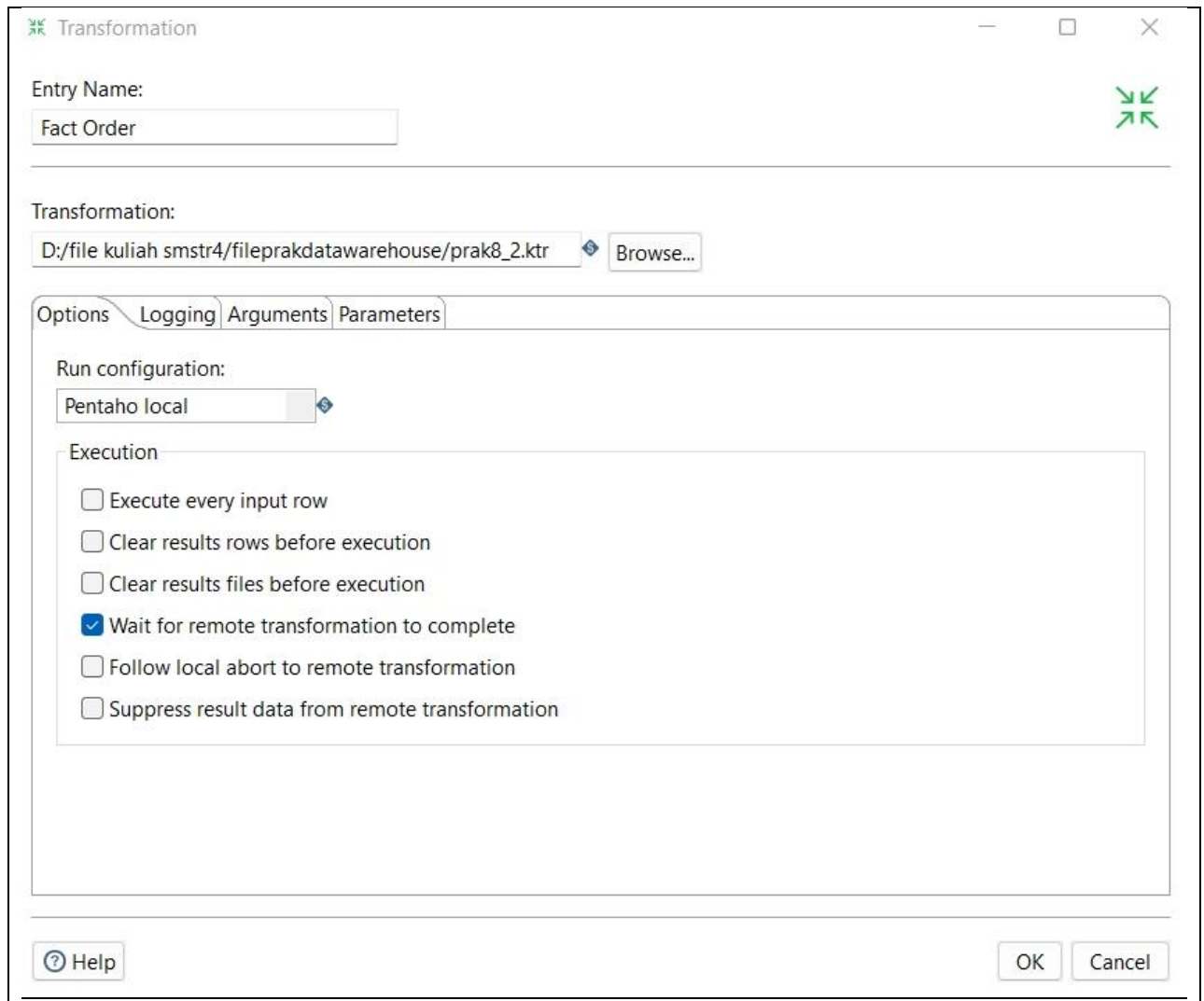


The image shows a 'Transformation' dialog box with the following fields and options:

- Entry Name:** Dim Customer
- Transformation:** \${Internal.Entry.Current.Directory}/prak 7/prak7_2.ktr (with a 'Browse...' button)
- Options** (selected tab), Logging, Arguments, Parameters
- Run configuration:** Pentaho local
- Execution** section with the following options:
 - ☐ Execute every input row
 - ☐ Clear results rows before execution
 - ☐ Clear results files before execution
 - ☒ Wait for remote transformation to complete
 - ☐ Follow local abort to remote transformation
 - ☐ Suppress result data from remote transformation
- Buttons:** Help, OK, Cancel

5. Fact Order – Transformation.





The screenshot shows the 'Transformation' configuration window in Pentaho. The 'Entry Name' field is set to 'Fact Order'. The 'Transformation' field shows the file path 'D:/file kuliah smstr4/fileprakdatawarehouse/prak8_2.ktr' with a 'Browse...' button. Below this is a tabbed interface with 'Options', 'Logging', 'Arguments', and 'Parameters' tabs. The 'Options' tab is active, showing 'Run configuration' set to 'Pentaho local'. Under the 'Execution' section, there are several checkboxes: 'Execute every input row' (unchecked), 'Clear results rows before execution' (unchecked), 'Clear results files before execution' (unchecked), 'Wait for remote transformation to complete' (checked), 'Follow local abort to remote transformation' (unchecked), and 'Suppress result data from remote transformation' (unchecked). At the bottom, there are 'Help', 'OK', and 'Cancel' buttons.

Transformation

Entry Name:
Fact Order

Transformation:
D:/file kuliah smstr4/fileprakdatawarehouse/prak8_2.ktr Browse...

Options Logging Arguments Parameters

Run configuration:
Pentaho local

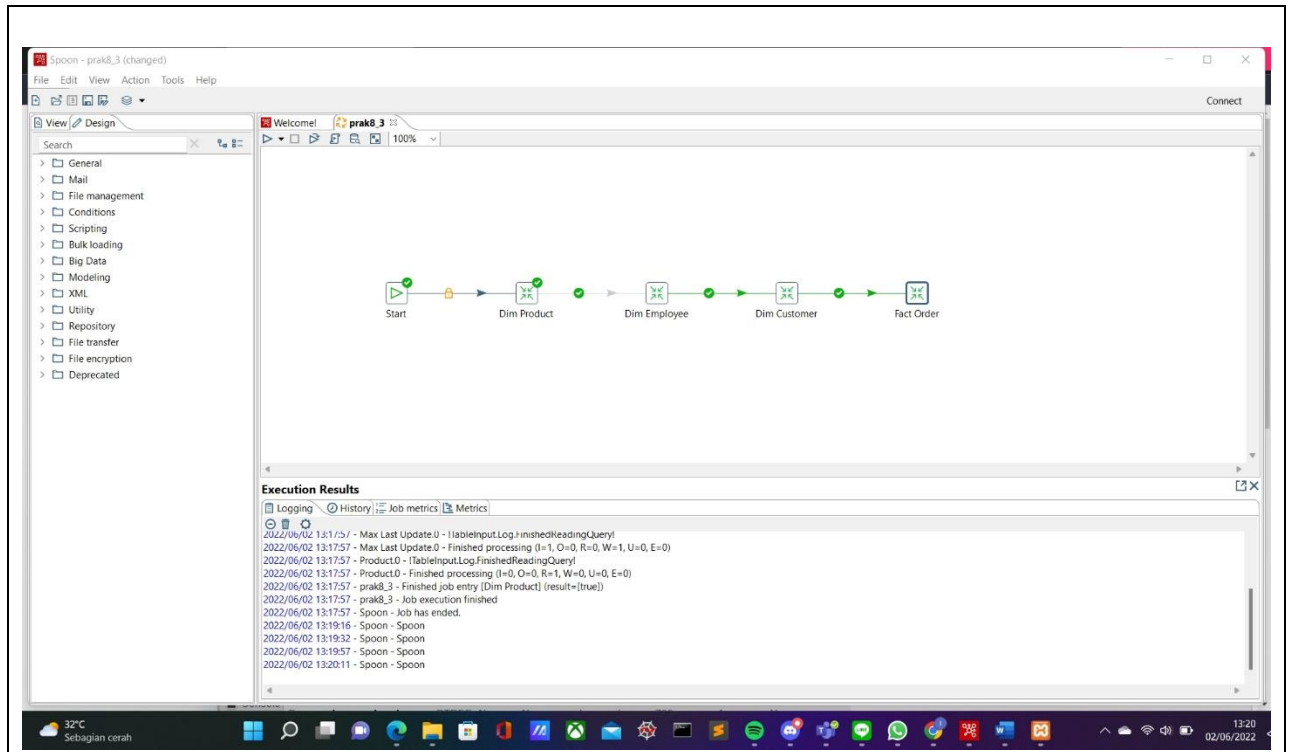
Execution

- ☐ Execute every input row
- ☐ Clear results rows before execution
- ☐ Clear results files before execution
- ☒ Wait for remote transformation to complete
- ☐ Follow local abort to remote transformation
- ☐ Suppress result data from remote transformation

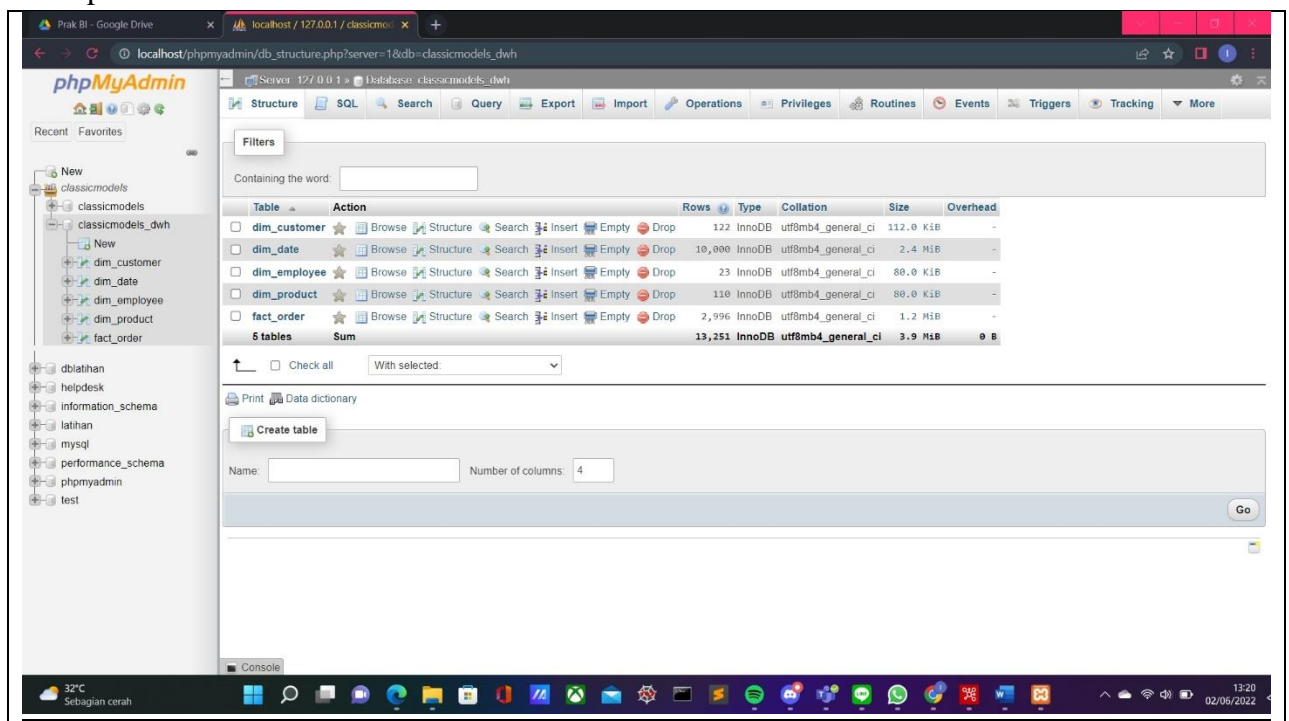
Help OK Cancel

6. Output





7. Output Pada Database



4. File Praktikum

Github Repository:

<https://github.com/IbnuFajar7/Data-Warehouse/tree/main/Prak-8>

5. Soal Latihan

Soal:

1. Apa perbedaan OLAP dengan OLTP?
2. Apa yang dimaksud dengan ETL?

Jawaban:

1. OLTP adalah pemrosesan transaksional sedangkan OLAP adalah sistem pemrosesan analitis. OLTP adalah sistem yang mengelola aplikasi berorientasi transaksi di internet misalnya ATM. OLAP adalah sistem online yang melaporkan ke kueri analitik multidimensi seperti pelaporan keuangan, peramalan, dll.
2. ETL adalah singkatan dari extract, transform, dan load. Melansir IBM, ia merupakan proses integrasi data. Di sana, data akan dikombinasikan dari berbagai sumber. Setelah itu, mereka disimpan di tempat bernama data warehouse.

6. Kesimpulan

- a. Dalam pengerjaan praktikum Data Warehouse, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- b. Kita dapat mengetahui bahwa setelah membuat dan menginput data yang begitu banyaknya kedalam sebuah database peer satu bagian, kemudian kita bisa menggabungkan data-data tersebut lagi menjadi suatu data yang lebih mudah dibaca dengan menggabungkan seluruh bagian menjadi satu bagian yang lebih besar lagi.

7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	✓	
2.	Latihan Kedua	✓	
3.	Latihan Ketiga	✓	

8. Formulir Umpan Balik



No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	20 Menit	1
2.	Latihan Kedua	20 Menit	1
3.	Latihan Ketiga	20 Menit	1

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang

